

Defending a Perimeter from a Ground Intruder Using an Aerial Defender: Theory and Practice

Elijah S. Lee¹, Daigo Shishika², Giuseppe Loianno³, and Vijay Kumar¹

Abstract—The perimeter defense game has received interest in recent years as a variant of the pursuit-evasion game. A number of previous works have solved this game to obtain the optimal strategies for defender and intruder, but the derived theory considers the players as point particles with first-order assumptions. In this work, we aim to apply the theory derived from the perimeter defense problem to robots with realistic models of actuation and sensing and observe performance discrepancy in relaxing the first-order assumptions. In particular, we focus on the hemisphere perimeter defense problem where a ground intruder tries to reach the base of a hemisphere while an aerial defender constrained to move on the hemisphere aims to capture the intruder. The transition from theory to practice is detailed, and the designed system is simulated in Gazebo. Two metrics for parametric analysis and comparative study are proposed to evaluate the performance discrepancy.

I. INTRODUCTION

The pursuit-evasion games (PEGs) have been widely investigated over the past years and are used in many applications including mobile robotics [1]. There are many variants of PEGs under different assumptions on the players and the environments. One route is to consider the players as point particles [2], [3]. Liang et al. [2] address a PEG with three point particles: target, attacker, and defender. The attacker aims to capture the target while avoiding the defender, and the defender aims to defend the target while trying to capture the attacker. Other work [3] allows the point particle players move in three dimensions and solves the differential games with three defenders and one intruder with equal speeds.

Researchers also have focused on solving the PEGs with real robots [4], [5], [6]. Vidal et al. [4] propose a hierarchical hybrid system to implement the pursuit-evasion game scenario on real UAVs and UGVs. The PEG between two pursuing and one evading unmanned aerial vehicles is solved in [5]. Deep learning based approach for vision-based UAV pursuit-evasion is implemented in [6].

This work formulates a variant of pursuit-evasion game known as the target-guarding problem [7]. In this problem, intruders aim to reach the target without being captured by defenders, while defenders try to capture intruders [2],

We gratefully acknowledge the support from ARL Grant DCIST CRA W911NF-17-2-0181, NSF Grant CNS-1521617, ARO Grant W911NF-13-1-0350, ONR Grants N00014-20-1-2822 and ONR grant N00014-20-S-B001, and Qualcomm Research.

¹The authors are with the GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104, USA. {elslee, kumar}@seas.upenn.edu

²The author is with George Mason University, Fairfax, VA 22030, USA. email: daigo.shishika@gmail.com

³The author is with the New York University, Tandon School of Engineering, Brooklyn, NY 11201, USA. email: loiannog@nyu.edu.

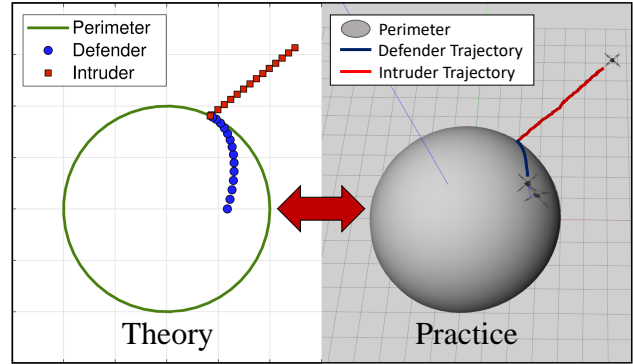


Fig. 1. Hemisphere perimeter defense game in theory and practice shows the discrepancy in executing optimal strategies.

[8], [9]. When defenders are constrained to move along the perimeter of the target region, we call this problem as perimeter defense game, which we refer to [10] for detailed survey. Previous works consider the players as point particles with first-order assumptions [2], [7], [8], [9], [10]. The optimal strategies are theoretically proved for multiplayer game on two-dimensional convex shapes [8] and for three-dimensional game on hemisphere [9].

There are many challenges in realizing the perimeter defense system with real robots, and the major challenges lie in coping with discrepancy between first-order assumptions of point particles and dynamics of real robots, as shown in Fig. 1. This work simulates the perimeter defense based on Unmanned Aerial Vehicles (UAVs). UAVs are deployed in various space such as power plant [11], penstock [12], forest [13], or disaster sites [14], which are good perimeter defense applications.

This paper extends the previous work [9] on hemisphere defense to apply the derived theory from point particles to real robots. We observe any performance discrepancy in relaxing first-order assumptions and discuss how to reduce such discrepancy. The contributions of the paper are (i) realizing perimeter defense between aerial defender and ground intruder from theory to practice; and (ii) performing parametric analysis of system scales and comparative study of strategies to evaluate the performance discrepancy in relaxing first-order assumptions.

Section II formulates the problem. Section III addresses the transition from point particle to real robot. To evaluate the discrepancy, two metrics are proposed in Section IV. Section V provides experimental results, and Section VI concludes the paper.

II. PROBLEM FORMULATION AND PRELIMINARIES

Consider a hemisphere perimeter O with radius of R as shown in Fig. 2. The defender D is constrained to move on the hemisphere while the intruder A is constrained to move on the ground plane. The positions of the players in spherical coordinates are: $\mathbf{z}_D = [\psi_D, \phi_D, R]$ and $\mathbf{z}_A = [\psi_A, 0, r]$, where ψ and ϕ are the azimuth and elevation angles, which gives the relative position as: $\mathbf{z} \triangleq [\psi, \phi, r]$, where $\psi \triangleq \psi_A - \psi_D$ and $\phi \triangleq \phi_D$ (see Fig. 2). Without loss of generality, we assume the defender's maximum speed is 1. The intruder is assumed to have a maximum speed $v \leq 1$ (otherwise, intruder always has a strategy to win the game). We denote that the velocities of defender and intruder are v_D and v_A , respectively. The game ends at time t_f with intruder's win if $r(t_f) = R$ and $|\psi(t_f)| + |\phi_D(t_f)| > 0$, whereas it ends with defender's win if $\phi_D(t_f) = \psi(t_f) = 0$ and $r(t_f) > R$. We call t_f as the *terminal time*.

A. Optimal breaching point for point particle

Given $\mathbf{z}_D, \mathbf{z}_A$, we call *breaching point* as a point on the perimeter at which the intruder tries to reach the target, as shown B in Fig. 2. We call the azimuth angle that forms the breaching point as *breaching angle*, denoted by θ , and call the angle between $(\mathbf{z}_A - \mathbf{z}_B)$ and the tangent line at B as *approach angle*, denoted by β .

It is proved in [9] that given the current positions of defender \mathbf{z}_D and intruder \mathbf{z}_A as point particles, there exists a unique breaching point such that the optimal strategy for both defender and intruder is to move towards it, known as *optimal breaching point*. The breaching angle and approach angle corresponding to the optimal breaching point are known as *optimal breaching angle*, denoted by θ^* , and *optimal approach angle*, denoted by β^* .

As stated in [9], although there exists no closed form solution for θ^* and β^* , they can be computed at any time by solving two governing equations:

$$\beta^* = \cos^{-1} \left(v \frac{\cos \phi_D \sin \theta^*}{\sqrt{1 - \cos^2 \phi_D \cos^2 \theta^*}} \right) \quad (1)$$

and

$$\theta^* = \psi - \beta^* + \cos^{-1} \left(\frac{\cos \beta^*}{r} \right) \quad (2)$$

B. Target time and payoff function

We call the *target time* as the time to reach B and define $\tau_D(\mathbf{z}_D, \mathbf{z}_B)$ as the *defender target time*, $\tau_A(\mathbf{z}_A, \mathbf{z}_B)$ as the *intruder target time*, and the following as *payoff* function:

$$p(\mathbf{z}_D, \mathbf{z}_A, \mathbf{z}_B) = \tau_D(\mathbf{z}_D, \mathbf{z}_B) - \tau_A(\mathbf{z}_A, \mathbf{z}_B) \quad (3)$$

The defender reaches B faster if $p < 0$ and the intruder reaches B faster if $p > 0$. Thus, the defender aims to minimize p while the intruder aims to maximize it.

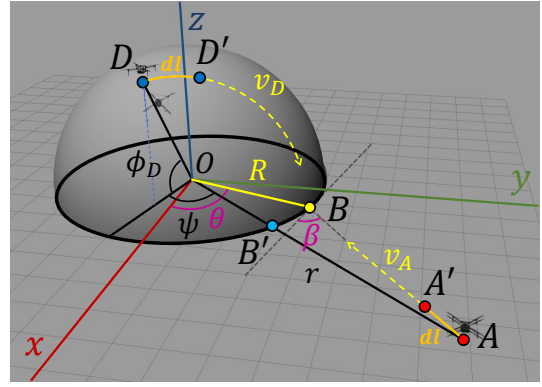


Fig. 2. The coordinate system and relevant variables.

C. Optimal strategies and Nash equilibrium

It is proven in [9] that the optimal strategies for both defender and intruder are to move towards the optimal breaching point at their maximum speed at any time.

Let Ω and Γ be the continuous v_D and v_A that lead to B so that $\tau_D(\mathbf{z}_D, \Omega) \triangleq \tau_D(\mathbf{z}_D, \mathbf{z}_B)$ and $\tau_A(\mathbf{z}_A, \Gamma) \triangleq \tau_A(\mathbf{z}_A, \mathbf{z}_B)$, and let Ω^* and Γ^* be the optimal strategies that minimize $\tau_D(\mathbf{z}_D, \Omega)$ and $\tau_A(\mathbf{z}_A, \Gamma)$, respectively, then the optimality in the game is given as a Nash equilibrium:

$$p(\mathbf{z}_D, \mathbf{z}_A, \Omega^*, \Gamma) \leq p(\mathbf{z}_D, \mathbf{z}_A, \Omega^*, \Gamma^*) \leq p(\mathbf{z}_D, \mathbf{z}_A, \Omega, \Gamma^*) \quad (4)$$

III. FROM THEORY TO PRACTICE

This section discusses the transition from theory to practice in executing the optimal strategies for hemisphere perimeter defense game.

A. From point particle to three-dimensional robot

The major challenge in bringing the theory closer to practice lies in representing the agents as three-dimensional robot. Previous works in perimeter defense focused on the point particle [2], [7], [8], [9], [10] to represent the defender and intruder, and the following are the assumptions made for the point particle, which may not hold true in working with the three-dimensional robot:

- It has no volume and thus it is scale invariant
- It moves with desired velocity instantly and precisely
- If it moves at its maximum speed, the speed is consistent along the trajectory
- It can accurately detect other agent's positions and react to it simultaneously
- Optimal trajectory obeys first-order assumptions

This work aims to simulate robots using the optimal strategies derived from theory, observe any discrepancy between the performances from theory and practice, and discuss how relaxing aforementioned assumptions would lead to the discrepancy. The employed robot is an UAV in Fig. 3. It has a dimension of 735mm \times 735mm \times 200mm and the mass of 1.8kg. This UAV well represents a three-dimensional robot relaxing the point particle assumptions.

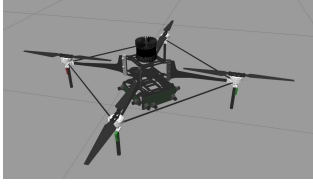


Fig. 3. An UAV carrying stereo cameras and Ouster lidar

B. Optimal breaching point in practice

In practice, we relax the assumption of scale invariance. The radius of hemisphere becomes a relevant variable R , as denoted in Fig. 2. Accordingly, (2) in practice becomes

$$\theta^* = \psi - \beta^* + \cos^{-1}\left(\frac{R \cos \beta^*}{r}\right) \quad (5)$$

although (1) stays the same in practice since all the parameters are scale invariant. We can solve for β^* and θ^* by solving two governing equations (1) and (5), and obtain the corresponding optimal breaching point in practice.

C. Controller

As mentioned in the previous assumptions [9], the controller used for theoretical approach was simply perfect. Both defender and intruder reacted to each other without any delay, and desired dynamics were enforced precisely and instantly to result in continuous optimal trajectories as shown in Fig. 1. The agents also identify each other's states without any error at any time (e.g. perfect state estimation).

The overall controller for practical approach is illustrated in Fig. 4. All modules represent the nodes in ROS and are connected by rostopics for real-time implementation. For each agent, the strategy module subscribes to the ground truth odometry that are published by both UAVs to calculate position command. In this way, state estimation is based on ground truth and becomes highly accurate. Then, the position command is fed into the SO3-based controller based on [15] to generate SO3 command. The SO3 commands consist of thrust and moment that

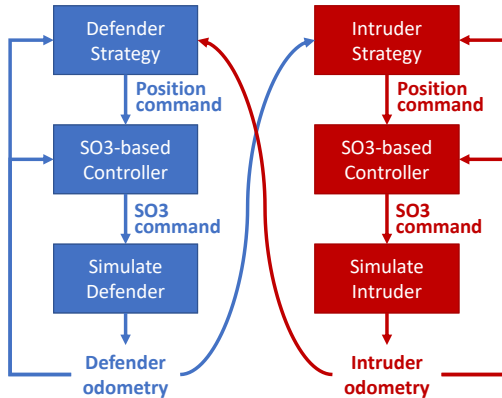


Fig. 4. Overall controller for cross feedback system

control the robot in low level. Finally, the simulated robot publishes the ground truth odometry to be fed into the SO3-based controller for closed loop control. Note that the defender and intruder strategy modules take in both agents' odometry. For this cross feedback system, the two odometry data are stored into queues and used for synchronized computation of the optimal strategy.

D. Executing optimal strategies

As stated in Sec.II-C, both defender and intruder move at their maximum speeds in executing optimal strategies in theory. It is also worth noting that the optimal trajectories are continuously changing based on the configuration of agents at given time because the optimal breaching point is newly calculated at any time to reflect the change in positions of the defender and intruder.

For this reactive and adversarial environment, we enforce the players to follow an infinitesimal segment of computed optimal trajectory at any time to maintain consistent maximum speeds. In Fig. 2, given the optimal breaching point B and current configurations of defender and intruder $\mathbf{z}_D = [\psi_D, \phi_D, R]$ and $\mathbf{z}_A = [\psi_A, 0, r]$, their current optimal trajectories are \overline{DB} and \overline{AB} , respectively. Based on the movements of defender and intruder, the optimal trajectories as well as the optimal breaching point may change over time, so we set the goal trajectory $\overline{DD'}$ and $\overline{AA'}$ as segments of original trajectories by taking the length of dl and dl' for defender and intruder, respectively. We give position command towards the end point of the infinitesimal segment, which can be viewed as controlling an instantaneous velocity of agents.

First, we aim to compute the position of D' $[x_D, y_D, z_D]$ after infinitesimal defender movement dl . For simplicity, assume D is on the zx -plane (i.e. $\psi_D = 0$) by rotating the hemisphere by $-\psi_D$ about the z -axis, and we will rotate the coordinates by ψ_D back to the original orientation at the end. Given the radius R , azimuth angle ϕ_D , and dl , we know the following: (i) D is on the hemisphere O with a center on the origin and a radius of R ; (ii) The arc distance between D and D' is dl ; (iii) D' is on the plane OBD . The conditions (i) and (ii) give

$$x_D^2 + y_D^2 + z_D^2 = R^2 \quad \text{and} \quad (6)$$

$$(x_D - R \cos \phi_D)^2 + y_D^2 + (z_D - R \sin \phi_D)^2 = dl^2 \quad (7)$$

We know the equation of a plane is given by

$$ax + by + cz = d \quad (8)$$

Therefore, the plane OBD is uniquely determined by three points $O(0, 0, 0)$, $B(R \cos \theta, R \sin \theta, 0)$, and $D(R \cos \phi_D, 0, R \sin \phi_D)$. With the condition (iii), (8) becomes

$$(\sin \phi_D \sin \theta) x_D - (\sin \phi_D \cos \theta) y_D - (\cos \phi_D \sin \theta) z_D = 0 \quad (9)$$

Together with (6), (7) and (9), we get

$$\begin{aligned} x_D &= R \cos \phi_D - \frac{dl^2 \cos \phi_D}{2R} + \frac{dl \sin^2 \phi_D \cos \theta}{2R} \cdot T \\ y_D &= \frac{dl \sin \theta}{2R} \cdot T \\ z_D &= R \sin \phi_D - \frac{dl^2 \sin \phi_D}{2R} - \frac{dl \sin \phi_D \cos \theta \cos \phi_D}{2R} \cdot T \end{aligned} \quad (10)$$

where

$$T = \sqrt{\frac{4R^2 - dl^2}{1 - \cos^2 \phi_D \cos^2 \theta}}$$

The final computation of the position x_D, y_D, z_D is summarized in Algorithm 1. Notice that line 4 and 5 rotate the system back to the original configuration. Similarly, the position of $A'[x_A, y_A, z_A]$ after infinitesimal intruder movement dl' is computed in Algorithm 2, and overall agents' strategies are summarized in Algorithm 3.

IV. DISCREPANCY ANALYSIS

In this section, we propose two metrics to evaluate the discrepancy between the performances from theory and practice. One metric measures the discrepancy from relaxing the scale invariant assumption made for point particle. This metric is crucial since the discrepancy between small and large scales would not allow scalability, which is essential for real-world applications. The other metric investigates if the traversed trajectory is the same as the optimal trajectory derived from theory by comparing its performances with the performance of other

Algorithm 1 $[x_D, y_D, z_D] = \text{MoveDefender}(\psi_D, \phi_D, \theta^*)$

Input: ψ_D, ϕ_D, θ^*

Output: Updated defender positions x_D, y_D, z_D

- 1: Set defender's small movement dl
 - 2: $\theta = \theta^* - \psi_D$
 - 3: Evaluate x_D, y_D, z_D using (10)
 - 4: $x_D = x_D \cos \psi_D - y_D \sin \psi_D$
 - 5: $y_D = x_D \sin \psi_D + y_D \cos \psi_D$
-

Algorithm 2 $[x_A, y_A, z_A] = \text{MoveIntruder}(x_A, y_A, z_A, \theta^*)$

Input: x_A, y_A, z_A, θ^*

Output: Updated intruder positions x_A, y_A, z_A

- 1: Set intruder's small movement dl'
 - 2: $x_B = R \cos \theta^*$
 - 3: $y_B = R \sin \theta^*$
 - 4: $l = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$
 - 5: $x_A = x_A + (x_B - x_A)dl'/l$
 - 6: $y_A = y_A + (y_B - y_A)dl'/l$
-

Algorithm 3 Defender/Intruder Strategy

Input: Agent odometry $x_D, y_D, z_D, x_A, y_A, z_A$

Output: Position command (updated x_D, y_D, z_D for defender and updated x_A, y_A, z_A for intruder)

- 1: Initialize $r, \psi_A, \psi_D, \phi_D$ using odometry
 - 2: **if** $\phi_D > 0$ or $|\psi_A - \psi_D| > 0$ **then**
 - 3: Compute optimal breaching angle θ^*
 - 4: **if** Defender Strategy **then**
 - 5: $[x_D, y_D, z_D] = \text{MoveDefender}(\psi_D, \phi_D, \theta^*)$
 - 6: **else if** Intruder Strategy **then**
 - 7: $[x_A, y_A, z_A] = \text{MoveIntruder}(x_A, y_A, z_A, \theta^*)$
 - 8: **end if**
 - 9: **end if**
-

baseline strategy. We chose these metrics to evaluate the discrepancy because other point particle assumptions (e.g. instant velocity change, simultaneous detection and reaction, and no fluctuation in maximum speed) highly depend on the hardware. A robot with better computing power will be closer to meet these assumptions, so we focus on the discrepancy in system scales and strategies.

A. Discrepancy in system scales

Fig. 5 shows different scales of environment (e.g. hemisphere) for perimeter defense game. Since the robot size is fixed, this variance may impact the dynamics of the UAV. For instance, if the robot traverses along the surface of small hemisphere, it may enforce the controller to execute abrupt turns and the resulting trajectory may not exactly follow the hemisphere surface.

To account for such performance discrepancy, we propose a metric l_d as shown in Fig. 6. This metric computes the geodesic between D' and A' , which represent the positions of defender and intruder at terminal time t_f , respectively. We purposely set the game for the intruder to win (e.g. intruder reaches to perimeter before defender does) because more interesting dynamics are happening for defender so we fix the behavior of intruder and evaluate l_d that is normalized by radius for different set of system scales. We also compare l_d with the outcome derived from first-order assumptions and observe how the relaxed point particle assumptions impact l_d , which would be closer to the theoretical outcome in ideal conditions.

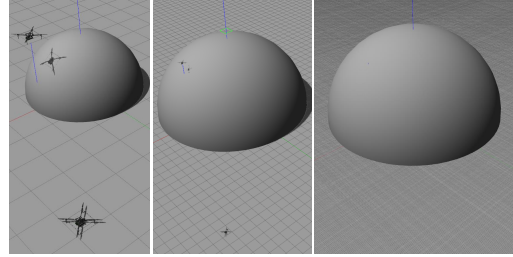


Fig. 5. Different system scales of environment and UAVs with a fixed scale. **Left:** Radius of 3. **Middle:** Radius of 10. **Right:** Radius of 100.

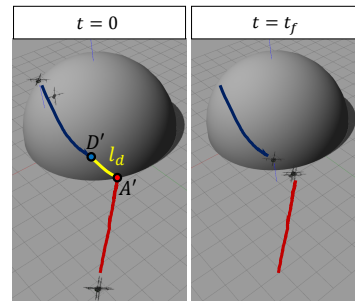


Fig. 6. Game setting for computing l_d . **Left:** Initial configuration when the game begins. **Right:** Final configuration at terminal time t_f .

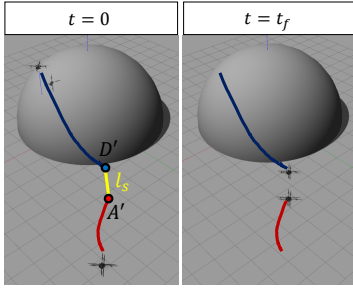


Fig. 7. Game setting for computing l_s . **Left:** Initial configuration when the game begins. **Right:** Final configuration at terminal time t_f .

B. Discrepancy in strategies

We define a metric l_s , shown as l_s in Fig. 7, to understand the discrepancy in different defender strategies. This metric measures the distance between D' and A' at terminal time t_f , which represents the distance between the agents at the end of the game. To obtain this metric, we intentionally set the game for the defender to win so that D' lies on the base of the hemisphere. In this manner, any plausible defender strategy can be evaluated because the game ends only if the defender finishes executing its strategy. If l_d is to be used as the metric, the game would end before defender fully executes the strategy, which makes it challenging to compare the performance.

In comparing strategies, we run two defender strategies: (1) optimal strategy that moves defender towards the breaching point B at its maximum speed at any time; and (2) baseline strategy that moves defender at its maximum speed at any time towards the azimuth angle of intruder, which is towards B' that is an intersection of \overline{OA} and the base of hemisphere as shown in Fig. 2. A strategy with larger l_s outperforms the other since it secures more time to defend the intruder.

V. EXPERIMENTS

This section evaluates the performance discrepancy between theory and practice based on system scales and strategies. We run all experiments using Gazebo with ROS.

A. Parametric analysis of system scales

In the parametric analysis, we vary the system scales by changing the radius of the hemisphere. A total of 22 different radii from 3 to 300 are used, and multiple experiments are run for each radius to compute l_d . Given radius R , the experiments are run with an initial configuration $\mathbf{z} = [\psi, \phi, r] = [0.9, 0.3\pi, 2R]$, infinitesimal movement $dl = dl' = 0.72$ (lead to $v_D = v_A = 0.8$), and $v = 1$.

Fig. 8 shows the change in l_d as a function of radius. l_d is normalized by radius so it should be consistent in theory regardless of system scales. It can be seen that l_d stays relatively small in the regime of small radius (e.g. $3 \leq R \leq 50$) and converges to the value derived from first-order assumptions as R increases. The error bar indicates that the standard deviation decreases as R increases. Fig.

9 confirms that small radius regime may result in non-smooth trajectories and the UAV trajectories converge to first-order assumptions as R increases. We infer that in the small radius regime, UAV dynamics is sensitive to control inputs since the scale of robot is comparable to the scale of environment. The UAV may accelerate well to get up to the maximum speed so l_d is lower and inconsistent with the outcome from the first-order assumptions.

In this way, the point particle assumption that agents move with desired velocity instantly is much relaxed. Accordingly, to reduce the discrepancy between theory and practice, a larger scale of environment relative to that of robot is preferred, and sensitivity analysis for the system would help the robot better obey the scale invariant assumptions.

B. Comparative study of strategies

For this study, we compute l_s for optimal and baseline strategies. Seven trials with different configuration are conducted to observe the performance discrepancy. An infinitesimal defender movement $dl = 1.36$ (lead to $v_D = 1.0$), an infinitesimal intruder movement $dl' = 0.36$ (lead to $v_A = 0.6$), and $v = 1$ are used to guarantee that the defender reaches to the base of hemisphere earlier than the intruder does.

Fig. 10 shows the variations of l_s normalized by radius for the two strategies in different trials. Although the discrepancy is very small, it is worth noting that the baseline strategy sometimes outperforms the optimal strategy. This tendency supports the claim that there exists discrepancy between theory and practice and that the optimal strategy derived from first-order assumptions could perform worse than other strategies in high-order dynamics. This discrepancy is due to the delay in executing the optimal strategy because computing the optimal breaching point takes some time. As a reference, one cycle of the cross feedback system in Fig. 4 takes about 70 msec, and the optimal breaching point calculation takes the major portion of it, which is about 10 msec. To compensate for the delay, robots with higher computing power can be employed.

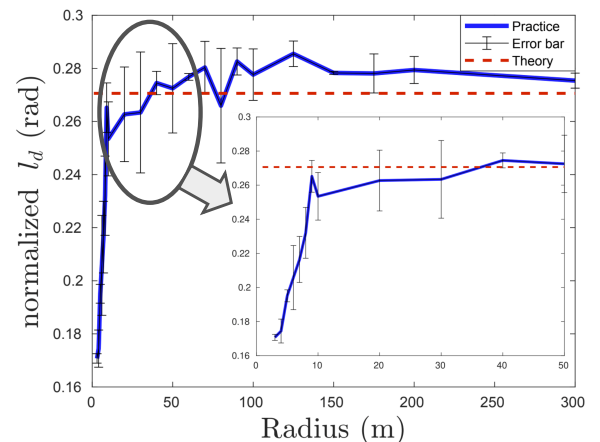


Fig. 8. Parametric analysis. Magnified view shows small radius regime.

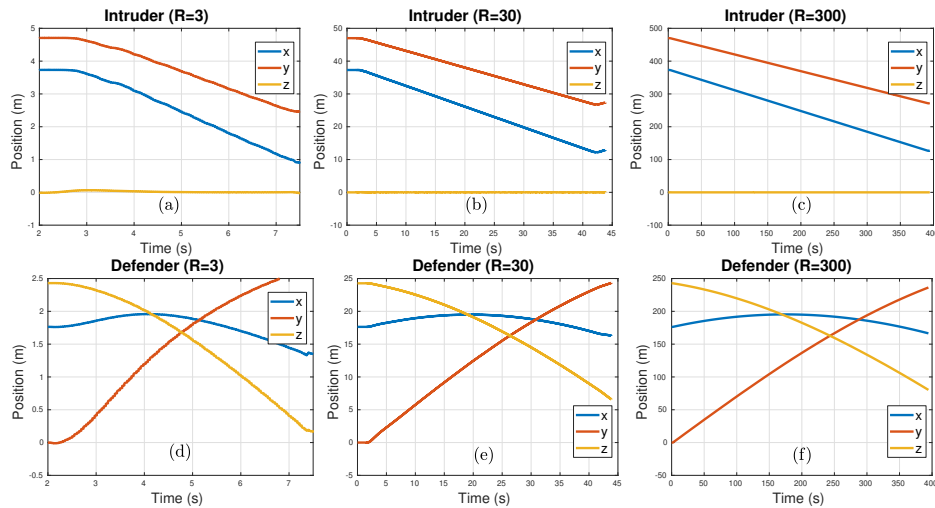


Fig. 9. The xyz positions of the defender and intruder with three set of radii. UAV trajectories converge to first-order assumptions as R increases.

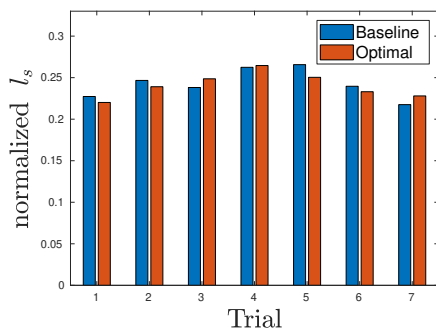


Fig. 10. Comparative study of baseline and optimal strategy.

VI. CONCLUSION

This paper aims to apply the theory with first-order assumptions derived from the hemisphere perimeter defense game to robots with realistic models, and observe performance discrepancy in relaxing point particle assumptions. We study the transition from theoretical point particles to practical three-dimensional robots using UAV models in Gazebo. To evaluate the performance discrepancy, two analyses are conducted. In the parametric analysis, it is found that UAV trajectories converge to first-order assumptions as the radius of the hemisphere increases. The comparative study shows that the optimal strategy derived from first-order assumptions can perform worse than other strategies in high-order dynamics due to the delay in executing the strategy. To cope with this discrepancy, the future work will focus on changing the first-order model to better capture the high dynamics of the UAVs and design pursuit strategies for the new model.

REFERENCES

- [1] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299, 2011.
- [2] Li Liang, Fang Deng, Zhihong Peng, Xinxing Li, and Wenzhong Zha. A differential game for cooperative target defense. *Automatica*, 102:58–71, 2019.
- [3] Rui Yan, Zongying Shi, and Yisheng Zhong. Construction of the barrier for reach-avoid differential games in three-dimensional space with four equal-speed players. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4067–4072. IEEE, 2019.
- [4] Rene Vidal, Omid Shakernia, H Jin Kim, David Hynchul Shim, and Shankar Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE transactions on robotics and automation*, 18(5):662–669, 2002.
- [5] Alexander Alexopoulos, Tobias Schmidt, and Essameddin Badreddin. Cooperative pursue in pursuit-evasion games with unmanned aerial vehicles. In *2015 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4538–4543. IEEE, 2015.
- [6] Sébastien Arola and Moulay A Akhrouf. Uav pursuit-evasion using deep learning and search area proposal. In *2019 IEEE international conference on robotics and automation*. IEEE, 2019.
- [7] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.
- [8] Daigo Shishika and Vijay Kumar. Local-game decomposition for multiplayer perimeter-defense problem. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2093–2100. IEEE, 2018.
- [9] Elijah S Lee, Daigo Shishika, and Vijay Kumar. Perimeter-defense game between aerial defender and ground intruder. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1530–1536. IEEE, 2020.
- [10] Daigo Shishika and Vijay Kumar. A review of multi agent perimeter defense games. In *International Conference on Decision and Game Theory for Security*, pages 472–485. Springer, 2020.
- [11] Elijah S Lee, Giuseppe Loianno, Dinesh Thakur, and Vijay Kumar. Experimental evaluation and characterization of radioactive source effects on robot visual localization and mapping. *IEEE Robotics and Automation Letters*, 5(2):3259–3266, 2020.
- [12] Ty Nguyen, Shreyas S Shivakumar, Ian D Miller, James Keller, Elijah S Lee, Alex Zhou, Tolga Özslan, Giuseppe Loianno, Joseph H Harwood, Jennifer Wozencraft, et al. Mavnet: An effective semantic segmentation micro-network for mav-based tasks. *IEEE Robotics and Automation Letters*, 4(4):3908–3915, 2019.
- [13] Steven W Chen, Guilherme V Nardari, Elijah S Lee, Chao Qu, Xu Liu, Roseli Ap Francelin Romero, and Vijay Kumar. Sloam: Semantic lidar odometry and mapping for forest inventory. *IEEE Robotics and Automation Letters*, 5(2):612–619, 2020.
- [14] Seoungjun Lee, Dongsoo Har, and Dongsuk Kum. Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion. In *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pages 84–89. IEEE, 2016.
- [15] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.